

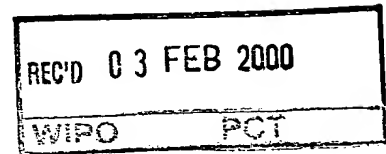
EU

De 99/3955



09/868239

Bescheinigung



Die Siemens Aktiengesellschaft in München/Deutschland hat eine Patentanmeldung unter der Bezeichnung

"Verfahren und Anordnung zur Vorhersage von Meßdaten
anhand vorgegebener Meßdaten"

am 16. Dezember 1998 beim Deutschen Patent- und Markenamt eingereicht.

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

Die Anmeldung hat im Deutschen Patent- und Markenamt vorläufig die Symbole G 06 F, G 01 D und G 05 B der Internationalen Patentklassifikation erhalten.

München, den 24. Januar 2000

Deutsches Patent- und Markenamt

Der Präsident

Im Auftrag

Weihmayr

Zeichen: 198 58 093.2



**PRIORITY
DOCUMENT**

SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

100-111780-100

THIS PAGE BLANK

Beschreibung**Verfahren und Anordnung zur Vorhersage von Meßdaten anhand vorgegebener Meßdaten**

5

Die Erfindung betrifft ein Verfahren und eine Anordnung zur Vorhersage von Meßdaten anhand vorgegebener Meßdaten.

10

Ein technisches System wirft oftmals die Frage nach einer Prognose anhand bekannter (Meß-)Daten auf, insbesondere im Hinblick auf eine Fehleranfälligkeit oder eine Kostenabschätzung.

15

Übernimmt eine solche Prognose ein Experte, so ist diese Prognose zumeist fehlerbehaftet. Eine exakte Bestimmung kann der Experte, zumindest für ein hinreichend komplexes System, nicht vornehmen.

20

Ein stochastischer Punktprozeß, insbesondere ein Poisson-Prozeß, ist aus [1] bekannt.

Die **Aufgabe** der Erfindung besteht darin, automatisch eine Vorhersage (Prognose) von Meßdaten anhand vorgegebener Meßdaten zu ermöglichen.

Diese Aufgabe wird gemäß den Merkmalen der unabhängigen Patentansprüche gelöst. Weiterbildungen der Erfindung ergeben sich auch aus den abhängigen Ansprüchen.

30

Zur Lösung der Aufgabe wird ein Verfahren zur Vorhersage von Meßdaten anhand vorgegebener Meßdaten angegeben, bei dem ein stochastischer Prozeß an die vorgegebenen Meßdaten angepaßt wird. Ab einem vorgegebenen Zeitpunkt werden Simulationsläufe bis zu einem Endzeitpunkt durchgeführt. Für jeden

35

Simulationslauf werden die prognostizierten Meßdaten bestimmt. Für den Endzeitpunkt erfolgt die Vorhersage von

Meßdaten innerhalb eines Wertebereichs, der durch die prognostizierten Meßdaten bestimmt ist.

5 Eine Weiterbildung besteht darin, daß ein Konfidenzintervall für die Vorhersage von Meßdaten bestimmt wird, indem die $a\%$ kleinsten und die $b\%$ größten prognostizierten Meßdaten eliminiert werden. Insbesondere kann $a\% = b\%$ sein. Beispielsweise kann somit ein 95%-iges Konfidenzintervall bestimmt werden, indem die 2,5% kleinsten und die 2,5%
10 größten prognostizierten Meßdaten unberücksichtigt bleiben.

Ein Vorteil besteht darin, daß von einem vorgegebenen Zeitpunkt aus die Meßdaten mit einer in einem Konfidenzintervall liegenden Genauigkeit vorhergesagt
15 (prognostiziert) werden können. Dies ermöglicht bereits in einem frühen Stadium die Erkennung beispielsweise einer Erfüllbarkeit bzw. Nichterfüllbarkeit einer mit den Meßdaten verbundenen Aufgabe, so daß daran geeignete Maßnahmen geknüpft werden können, um einer prognostizierten
20 Nichterfüllung entgegenzuwirken. Dies ist insbesondere von Bedeutung bei einem komplexen System, z.B. einem Softwareentwicklungsprozeß, bei dem innerhalb einer späteren Testphase aufzeigbar ist, inwieweit eine Zeitplanung bis zur Fertigstellung der Software eingehalten werden kann. Umso
25 wichtiger ist es, hierbei frühzeitig einer deutlichen Verzögerung, beispielsweise in einer Integrationstest-Phase, entgegenwirken zu können. Dies wirkt sich zum einen auf die Erfüllbarkeit innerhalb der gesetzten Frist (Zeitrahmen), zum anderen direkt auf die Kosten aus, da eine Nichterfüllung in
30 der vereinbarten Zeit regelmäßig zusätzliche Kosten verursacht.

Eine Ausgestaltung besteht darin, daß der stochastische Prozeß ein nichthomogener Poisson-Prozeß ist.
35

Insbesondere ist es eine Ausgestaltung, daß die Meßdaten Anzahlen von Fehlern sind. Dies entspricht beispielsweise

- einer Softwareentwicklung, bei der abhängig von in einer Testphase gemessenen Fehlern ein Reifegrad derselben dokumentiert wird. Abhängig von diesem Reifegrad resultiert direkt die Fertigstellung, d.h. solange nicht ein Großteil der Fehler aus der Software entfernt wurde, kann diese nicht an Kunden ausgeliefert werden. Dies drückt sich insbesondere durch Aufwand (beim Testen und Korrigieren der Fehler) und Kosten (für die Zeitverzögerung bei der Auslieferung) aus.
- 10 Zur Lösung der Aufgabe wird auch ein Verfahren zur Vorhersage von Meßdaten anhand vorgegebener Meßdaten angegeben, bei dem ein stochastischer Prozeß an die vorgegebenen Meßdaten angepaßt wird. Es wird ein Intervall ermittelt, indem anhand des stochastischen Prozesses gewonnene
- 15 Wahrscheinlichkeitswerte der Größe nach um einen Erwartungswert sortiert werden. Die Vorhersage von Meßdaten erfolgt durch Orientierung an dem Intervall, insbesondere an den Wahrscheinlichkeitswerten innerhalb des Intervalls.
- 20 Eine Weiterbildung besteht darin, daß die anhand des stochastischen Prozesses gewonnenen Wahrscheinlichkeitswerte der Größe nach symmetrisch um den Erwartungswert sortiert werden. Damit ist insbesondere gemeint, daß der größte Wahrscheinlichkeitswert die Mitte des Intervalls, also den Erwartungswert kennzeichnet, wohingegen der nächst größere Wahrscheinlichkeitswert rechts- bzw. linksseitig des Erwartungswertes angeordnet wird. Der nachfolgend nächst höhere Wahrscheinlichkeitswert wird gegenüber auf der anderen Seite des Erwartungswertes symmetrisch angeordnet.
- 30 Auf diesem Wege erhält man analytisch (konstruktiv) ein Intervall, das wiederum durch seine Breite angibt, welche Wahrscheinlichkeitswerte für die Vorhersage der Meßdaten eine Rolle spielen.
- 35 Insbesondere ist es eine Ausgestaltung, daß die Breite des Intervalls bestimmt wird, indem diejenigen

Wahrscheinlichkeitswerte unberücksichtigt bleiben, die unterhalb einer vorgegebenen Schranke liegen.

5 Dadurch ergibt sich ein Intervall (Konfidenzintervall),
welches durch die Schranke eine bestimmte Breite aufweist.
Diese Breite entspricht der Sicherheit der Vorhersage von
Meßdaten.

10 Geht man davon aus, daß der stochastische Prozeß ein
nichthomogener Poisson-Prozeß ist, so wird insbesondere auf
einer Zeitachse t durch den nichthomogenen Poisson-Prozeß
eine Schrittweite bestimmt, die angibt, wann der nächste
Fehler auftritt. Durch die Eigenschaft der
15 Gedächtnislosigkeit des nichthomogenen Poisson-Prozesses wird
von jedem zu einem bestimmten Zeitpunkt aufgetretenen Fehler
"gedächtnislos" nach einem für den nächsten Fehler
kennzeichnenden Zeitpunkt gesucht.

20 Auch wird zur Lösung der Aufgabe eine Anordnung zur
Vorhersage von Meßdaten anhand vorgegebener Meßdaten
angegeben, bei der eine Prozessoreinheit vorgesehen ist, die
derart eingerichtet ist, daß

- a) ein stochastischer Prozeß an die vorgegebenen Meßdaten
anpaßbar ist;
- 25 b) ab einem vorgegebenen Zeitpunkt Simulationsläufe bis
zu einem Endzeitpunkt durchführbar sind;
- c) für jeden Simulationslauf die prognostizierten
Meßdaten ermittelbar sind;
- 30 d) für den Endzeitpunkt die Vorhersage von Meßdaten
innerhalb eines Wertebereichs, der durch die
prognostizierten Meßdaten bestimmt ist, vorhersagbar
ist.

35 Ferner wird zur Lösung der Aufgabe eine Anordnung zur
Vorhersage von Meßdaten anhand vorgegebener Meßdaten
angegeben, bei der eine Prozessoreinheit vorgesehen ist, die
derart eingerichtet ist, daß

- a) ein stochastischer Prozeß an die vorgegebenen Meßdaten anpaßbar ist;
- b) ein Intervall ermittelbar ist, indem anhand des stochastischen Prozesses gewonnene Wahrscheinlichkeitswerte der Größe nach um einen Erwartungswert sortiert werden;
- c) die Vorhersage von Meßdaten innerhalb der Grenzen des Intervalls erfolgt.

Die Anordnungen sind insbesondere geeignet zur Durchführung der erfindungsgemäßen Verfahren oder der vorstehend erläuterten Weiterbildungen.

Ausführungsbeispiele der Erfindung werden nachfolgend anhand der Zeichnung dargestellt und erläutert.

Es zeigen

Fig.1 eine Skizze, die eine akkumulierte Fehleranzahl über einem Testzeitraum darstellt;

Fig.2 eine Skizze, die überlagerte Konfidenzintervalle für verschiedene Prozeßmodelle darstellt;

Fig.3 ein Flußdiagramm mit Schritten eines Verfahrens zur Vorhersage von Meßdaten anhand vorgegebener Meßdaten;

Fig.4 ein weiteres Flußdiagramm mit Schritten eines Verfahrens zur Vorhersage von Meßdaten anhand vorgegebener Meßdaten;

Fig.5 eine Prozessoreinheit;

Um eine zu erwartende Fehleranzahl in einem technischen Prozeß, beispielsweise in einem Software-Entwicklungsprozeß, prognostizieren zu können, werden nichthomogene Poisson-

Prozesse (NHPP) kalibriert, d.h. an Meßdaten, z.B. das Auftreten der Fehler über die Zeit, angepaßt. Mit

$$\{N(t)\}_{t \in \mathbb{R}^+} \quad (1)$$

5

wird ein zu dem stochastischen Punktprozeß (nichthomogenen Poisson-Prozeß) assoziierter Zählprozeß und mit einem Zeitpunkt t_0 ein Ende eines Testzeitraumes, also ein Zeitpunkt an dem die vorgegebenen Daten enden, bezeichnet. Es werden die stochastischen Prozesse

10

$$\{U(t)\}_{t \in \mathbb{R}^+} \quad \text{und} \quad (2)$$

$$\{O(t)\}_{t \in \mathbb{R}^+} \quad (3)$$

15

gesucht mit

$$P(U(t) \leq N(t) - N(t_0) \leq O(t) - N(t_0) = n_0) \geq \alpha \quad (4),$$

20

für alle Zeitpunkte $t > t_0$ und gegebene Werte $\alpha \in (0,1)$ (Konfidenzniveau) sowie $n_0 \in \mathbb{N}$. Nachfolgend werden insbesondere die Zuwächse des stochastischen Zählprozesse bezogen auf den Zeitpunkt t_0 betrachtet.

25

Für den hier vorliegenden Fall, in dem Gleichung (1) einen nichthomogenen Poisson-Prozeß bezeichnet, gilt (vgl. [1])

$$P(N(t_1) - N(t_0) = \ell) = \exp(-[i(t_1) - i(t_0)]) \cdot \frac{[i(t_1) - i(t_0)]^\ell}{\ell!} \quad (5)$$

30 für

$$0 \leq t_0 < t_1 < \infty, \ell \in \mathbb{N}_0 \quad (6)$$

und eine Intensität (englisch: intensity, mean measure, mean value function)

35

$$i: \mathbf{R}^+ \rightarrow \mathbf{R}^+, t \mapsto i(t) = EN(t) \quad (7).$$

Da gemäß der Eigenschaft des Poisson-Prozesses die Zuwächse
 5 (hier: Fehlerzuwächse) unabhängig von vorangegangenen
 Zuwächsen sind, kann Gleichung (5) für Zeitpunkte $t > t_0$ zur
 Bestimmung eines (möglichst kleinen) Intervalls

$$[g_u, g_o] \equiv [g_u(t), g_o(t)] \subset \mathbf{N}_0 \quad (8)$$

10

vereinfacht werden zu

$$\sum_{\ell=g_u}^{g_o} P(N(t) - N(t_0) = \ell) \geq \alpha \quad (9).$$

15 Aufgrund der Unimodalität der Poisson-Zähldichte kann ein
 Intervall $[g_u, g_o]$ wie folgt ermittelt werden:

Schritt 1: sortiere die Elementarwahrscheinlichkeiten

20

$$p_\ell := P(N(t) - N(t_0) = \ell), \ell \in \mathbf{N}_0$$

in absteigender Reihenfolge und bezeichne die
 derart sortierten Werte mit

25

$$p(0), p(1), \dots \quad (\text{d.h. } \{p_0, p_1, \dots\} = \{p(0), p(1), \dots\} \text{ und } p(0) \geq p(1) \geq \dots);$$

$$\text{Schritt 2: bestimme } \ell_{\min} := \min \left\{ \ell \in \mathbf{N}_0 \mid \sum_{i=0}^{\ell} p(i) \geq \alpha \right\};$$

30 Schritt 3: bestimme eine Indexmenge

$$I := \{i_0, \dots, i_{\ell_{\min}}\} \subset \mathbf{N}_0 \quad \text{mit}$$

$$\{p_{i_0}, \dots, p_{i_{\ell_{\min}}}\} = \{p(0), \dots, p(\ell_{\min})\};$$

Schritt 4: setze $g_u := \min_{i \in I} \{i\}$ und $g_o := \max_{i \in I} \{i\}$.

5

Das Intervall aus Gleichung (8) wird auch als Prognose-Intervall bezeichnet.

10 Stochastische Simulation (zweiter Ansatz)

Das beschriebene Konfidenzintervall kann simulativ bestimmt werden durch folgende Schritte:

15 Schritt 1: ~~starte~~ am Zeitpunkt t_0 der letzten Fehlermeldung ~~mit m voneinander unabhängigen Simulationsläufe,~~
~~die auf dem gewählten Prozeßmodell basieren;~~

20 Schritt 2: ~~beende~~ einen Simulationslauf, sobald der gewünschte Endzeitpunkt t_e erreicht ist;

Schritt 3: wiederhole Schritt 2 solange bis alle Simulationsläufe beendet sind;

25 Schritt 4: sortiere die Anzahlen $\hat{N}_i(t_e)$ der im i -ten Simulationslauf generierten Fehler in dem Zeitraum $(t_0, t_e]$, $i=1, \dots, m$, in absteigender Reihenfolge und bezeichne die derart sortierten Werte mit $\hat{N}_{(1)}(t_e), \dots, \hat{N}_{(m)}(t_e)$;

30

Schritt 5: setze

$$\hat{g}_u := \hat{N}_{(\lfloor m \cdot \alpha / 2 \rfloor)}(t_e) \text{ und}$$

$$\hat{g}_o := \hat{N}_{(\lceil m \cdot (1 - \alpha / 2) \rceil)}(t_e),$$

d.h. eliminiere die $(100 \cdot (1 - \alpha) / 2)\%$ kleinsten und größten Werte.

- 5 Daraus resultiert unmittelbar das Konfidenzintervall.

Jeder einzelne Simulationslauf basiert auf einem Simulationsalgorithmus, der aus (vgl. [2]) bekannt ist: Simulative Erzeugung von Zwischenankunftszeiten eines
10 nichthomogenen Poisson-Prozesses:

Schritt 1: Setze $\bar{\lambda} := \sup_{t \geq t_s} \{\lambda(t)\}$, wobei gilt:

$$\lambda(t) := \left. \frac{di}{dt} \right|_t \quad (10).$$

- 15 Schritt 2: erzeuge eine mit dem Parameter $\bar{\lambda}$ exponentialverteilte (Pseudo-)Zufallsvariable X, d.h. $X := -\log(U) / \bar{\lambda}$, wobei U auf (0,1) gleichverteilt ist.

- 20 Schritt 3: erzeuge eine auf (0,1) gleichverteilte Zufallsvariable U.

- 25 Schritt 4: falls $U \leq \lambda(t_s + X) / \bar{\lambda}$, dann setze $t^* := t_s + X$; andernfalls setze $t_s := t_s + X$ und gehe zu Schritt 1.

30 **Fig.1** zeigt beispielhaft eine Skizze, die eine akkumulierte Fehleranzahl über einem vorgegebenen Testzeitraum darstellt. Ab einem Zeitpunkt t_0 zeigt sich ein Vorhersageintervall für alle Zeitpunkte $t_0 + x$.

Allgemein ergibt sich die Ableitung der Intensität i gemäß Gleichung (10) zu λ . Beispielsweise ergibt sich:

10

$$a) \lambda(t) = a \cdot b \cdot c \cdot \exp(-bt^c) \cdot t^{c-1}$$

($\lambda(t)$ ist für $c \leq 1$ streng monoton fallend, für $c > 1$ unimodal mit einem eindeutigen Maximum an einer Stelle

$$t_{\max} = \sqrt[c]{\frac{c-1}{bc}}).$$

5

b) $\bar{\lambda}$ ergibt sich nach den obigen Überlegungen zu

$$\bar{\lambda} = \begin{cases} \lambda(t_s), & (c \leq 1) \vee (t_s \geq t_{\max}) \\ \lambda(t_{\max}), & \text{sonst.} \end{cases}$$

- 10 In **Fig.2** ist eine Skizze gezeigt, die überlagerte Konfidenzintervalle darstellt. Insbesondere wird dadurch ersichtlich, daß eine etwaige Prognose umso breiter streut, je weiter sie in die Zukunft reicht. Insbesondere können durch verschiedene Prozeßmodelle berechnete
- 15 Konfidenzintervalle auf die in Fig.2 dargestellte Art visualisiert werden.

- In **Fig.3** ist ein Flußdiagramm mit Schritten eines Verfahrens zur Vorhersage von Meßdaten anhand vorgegebener Meßdaten dargestellt. In einem Schritt 301 wird ein stochastischer Prozeß, insbesondere ein nichthomogener Poisson-Prozeß (als Vertreter für einen stochastischen Zählprozeß) an vorgegebenen Meßdaten angepaßt. In einem Schritt 302 werden Simulationsläufe durchgeführt, die ausgehend von dem
- 20 Zeitpunkt t_0 bis zu einem zu prognostizierenden Endzeitpunkt t_e laufen. Für jeden Simulationslauf werden in einem Schritt 303 prognostizierte Meßdaten bestimmt und eine Vorhersage von Meßdaten auf einen Bereich eingeschränkt, der von den durch die Simulationsläufe bestimmten Meßdaten abgedeckt wird
- 25 (siehe Schritt 304). In einem Schritt 305 wird ein Konfidenzintervall bestimmt, indem jeweils ein vorgegebener Teil größter und kleinster prognostizierter Meßdaten unberücksichtigt bleibt (dies entspricht besagtem Bereich). In einem Schritt 306 wird das Verfahren beendet.

Fig.4 zeigt ein weiteres Flußdiagramm mit Schritten eines Verfahrens zur Vorhersage von Meßdaten anhand vorgegebener Meßdaten. Dazu wird in einem Schritt 401 eine Anpassung eines stochastischen Prozesses, insbesondere eines nichthomogenen Poisson-Prozesses an die vorgegebenen Meßdaten durchgeführt. Anhand des stochastischen Prozesses werden Wahrscheinlichkeitswerte bestimmt, die der Größe nach um einen Erwartungswert sortiert werden (vergleiche Schritt 402). Durch diesen Sortiervorgang wird ein Intervall, hier ein Konfidenzintervall, bestimmt. Die Breite des Konfidenzintervalls ergibt sich durch Vergleich der akkumulierten Wahrscheinlichkeiten mit einer vorgegebenen Schwelle. Durch das Konfidenzintervall ist, wie oben erklärt wurde, eine Verteilung bzw. Unschärfe von einem Zeitpunkt t_0 in die Zukunft gegeben, die eine Abschätzung der Meßdaten in der Zukunft ermöglicht (siehe Schritt 403). In einem Schritt 404 wird das Verfahren beendet.

In **Fig.5** ist eine Prozessoreinheit PRZE dargestellt. Die Prozessoreinheit PRZE umfaßt einen Prozessor CPU, einen Speicher SPE und eine Input/Output-Schnittstelle IOS, die über ein Interface IFC auf unterschiedliche Art und Weise genutzt wird: Über eine Grafikschnittstelle wird eine Ausgabe auf einem Monitor MON sichtbar und/oder auf einem Drucker PRT ausgegeben. Eine Eingabe erfolgt über eine Maus MAS oder eine Tastatur TAST. Auch verfügt die Prozessoreinheit PRZE über einen Datenbus BUS, der die Verbindung von einem Speicher MEM, dem Prozessor CPU und der Input/Output-Schnittstelle IOS gewährleistet. Weiterhin sind an den Datenbus BUS zusätzliche Komponenten anschließbar, z.B. zusätzlicher Speicher, Datenspeicher (Festplatte) oder Scanner.

Nachfolgend werden ein Algorithmus zur Bestimmung von Konfidenzintervallen für Prognosen und ein Algorithmus zur simulativen Bestimmung von Konfidenzintervallen für Prognosen in der Notation der Programmiersprache C angegeben.

Programm 1:

```

/* Bestimmung von Konfidenzintervallen für Prognosen */
/* basierend auf dem verallgemeinerten Goel-Okamoto-Modell */
5  #include <stdlib.h>
   #include <math.h>
   #include <stdio.h>

   #define true 1
10  #define false -1

   double mv_genGO(double,double,double,double); double poisson(double,long); void ki_nhpp();
   int main(argc,argv)
   int argc;
15  char *argv[];
   {
   double a,b,c,bt,st,kn;
   long low,upp,lauf;

20  if (argc<7) {
   printf("\n\nZu wenig Argumente! \n\n");
   printf("Aufruf: %s <Par1> <Par2> <Par3> <Startzeit> <Endzeit>",
   "<KNiveau>\n\n", argv[0]); return 1;
   }

25  a = atof(argv[1]);
   b = atof(argv[2]);
   c = atof(argv[3]);
   bt= atof(argv[4]);
30  st= atof(argv[5]);
   kn= atof(argv[6]);

   for (lauf=1;lauf< ;lauf++) {
   ki_nhpp(mv_genGO,a,b,c,bt,bt+lauf*(st-bt)/10.,kn,&low,&upp);
35  printf("Zeitpunkt: %8.2f Fehlerintervall: [%d,%d]\n",
   bt+lauf*(st-bt)/10., low, upp);
   }
   return 0;
   }

40  double mv_genGO(x,a,b,c)
   double x,a,b,c;
   { return( a*(1.0-exp(-b*pow(x,c))) ); }

45  double poisson(lambda,wert)
   double lambda;
   long wert;
   {
   long i;
50  double itval,hv;

   if (lambda<600) {
   itval = exp(-lambda);
   for (i=wert;i>=1;i--) { itval *= lambda/(double)i; }
55  }

```

```

else {
    hv = xp(-lambda/(double)wert);
    itval = 1.0;
    for (i=wert;i>=1;i--) { itval *= lambda/(double)i*hv; }
5   }
    return ( itval );
}

void ki_nhpp(mv_nhpp, par1_nhpp, par2_nhpp, par3_nhpp,
10   start_time, stop_time, k_niveau, lower, upper) double mv_nhpp(double,double,double,double);
    double par1_nhpp, par2_nhpp, par3_nhpp, start_time, stop_time, k_niveau; long *lower, *upper;
    {
        long lauf;
        int lborder,mod_low,mod_upp;
15   double sum,tmp_mv, val_l, val_u;

        tmp_mv = mv_nhpp(stop_time,par1_nhpp,par2_nhpp,par3_nhpp) -
        mv_nhpp(start_time,par1_nhpp,par2_nhpp,par3_nhpp); lauf = (long)tmp_mv;
        *lower = lauf-1;
20   *upper = lauf+1; mod_low= false; mod_upp= false; sum = poisson(tmp_mv,lauf); val_l =
        poisson(tmp_mv,*lower); val_u = poisson(tmp_mv,*upper);
        while (sum<k_niveau) {
            if (val_l<val_u) {
                sum += val_u;
25   (*upper)++;
                lborder = false;
                mod_upp = true;
                val_u = poisson(tmp_mv,*upper);
            }
            else {
30   sum += val_l;
                (*lower)--;
                lborder = true;
                mod_low = true;
                val_l = poisson(tmp_mv,*lower);
35   }
            }
        }

        if (lborder == true) { (*lower)++; }
40   else { (*upper)--; }

        if (mod_low == false) { (*lower)++; }
        if (mod_upp == false) { (*upper)--; }

45   return;
    }

```

Programm 2:

```

/* Simulative Bestimmung von Konfidenzintervallen für Prognosen */
/* basierend auf dem verallgemeinerten Goel-Okamoto-Modell */
5  #include <stdlib.h>
   #include <math.h>
   #include <time.h>
   #include <stdio.h>
   #include <values.h>

10  #define true  1
   #define false -1

   double drand48(void);
15  void srand48(long);

   double sim_exp(double); double lambda_genGO(double,double,double,double); void sim_nhpp();
   int main(argc,argv)
   int argc;
20  char *argv[];
   {
   time_t t; double a,b,c,bt,st,pnt[1000000],check_time[12]; long lauf,no_pnt,seed_run; int clauf;
   FILE *datei;
   if (argc<6) {
25  printf("\n\nZuwenig Argumente!\n\n");
   printf("Aufruf: %s<Par1> <Par2> <Par3> <Startzeit> <Endzeit>\n\n",
   argv[0]); return 1;
   }

30  datei = fopen("sim:seed","r");
   if (datei==NULL) {
   seed_run = 1;
   }
   else {
35  fscanf(datei,"%6d",&seed_run); fclose(datei); seed_run++;
   }

   datei = fopen("sim.seed","w+");
   fprintf(datei, "%6d\n", seed_run );
40  fclose(datei);

   time (&t);          /* Initialisierung des */
   t += seed_run*100;   /* Zufallszahlengenerators */
   srand48 ((unsigned long) t); /* mit Hilfe der Systemzeit */
45  a = atof(argv[1]);
   b = atof(argv[2]);
   c = atof(argv[3]);
   bt= atof(argv[4]);
   st= atof(argv[5]);

50  sim_nhpp(lambda_genGO,a,b,c,bt,st,&pnt,&no_pnt);
   for (lauf=1;lauf<=no_pnt;lauf++) {
   printf("%15.7f %10d \n", pnt[lauf], lauf);
55  }

```


15

```

datei = fopen("ki.tmp","a");
for (lauf=1;lauf< ;lauf++) {
check_time[lauf] = bt+lauf*(st-bt)/10.;
}
5 check_time[11] = pnt[no_pnt]+1; /* größer als die größte
simulierte Zeit */
clauf = 1;
for (lauf=1;lauf<=no_pnt;lauf++) {
10 while (pnt[lauf]>=check_time[clauf]) { fprintf(datei, "%8.2f %6d ", check_time[clauf], lauf-1); clauf++;
}
}

if (pnt[no_pnt] < check_time[10]) {
15 for (lauf=clauf;lauf< ;lauf++) { fprintf(datei, "%8.2f %6d ", check_time[lauf], no_pnt);
}
}

fprintf(datei, "\n");
fclose(datei);

20 return 0;
}

double sim_exp(lambda)
25 double lambda;
{ return( -log(drand48())/lambda ); }

double lambda_genGO(x,a,b,c)
double x,a,b,c;
30 { return( a*b*c*pow(x,c-1)*exp(-b*pow(x,c)) ); }

void sim_nhpp(lambda_nhpp, par1_nhpp, par2_nhpp, par3_nhpp,
start_time, stop_time, path, no_points) double lambda_nhpp(double,double,double,double); double
par1_nhpp, par2_nhpp, par3_nhpp, start_time, stop_time; double path[]; long *no_points;
35 {
double sim_time,x,u,x_bar,lambda_bar;
*no_points=0;
sim_time = start_time;

40 do {
if (par3_nhpp<=1) { lambda_bar = lambda_nhpp(sim_time,par1_nhpp,par2_nhpp,par3_nhpp);
}
else {
x_bar = pow((par3_nhpp-1.0)/par2_nhpp/par3_nhpp,1.0/par3_nhpp);
45 if (sim_time>=x_bar) {
lambda_bar = lambda_nhpp(sim_time,par1_nhpp,par2_nhpp,par3_nhpp);
}
else {
lambda_bar = lambda_nhpp(x_bar,par1_nhpp,par2_nhpp,par3_nhpp);
50 }
}
}

x = sim_exp(lambda_bar);
55 u = drand48();

```

)

16

```
if (u<=lambda_nhpp(sim_time+x,par1_nhpp,par2_nhpp,par3_nhpp)/lambda_bar) { (*no_points)++;  
    path[*no_points]=sim_time+x;  
    }  
sim_time+=x;  
5  }  
while (sim_time<=stop_time);  
return;  
}
```

Programm 3:

```

/* Bestimmung der Konfidenzintervalle aus den Simulationsdaten */
/* (die Simulationsdaten werden in aufsteigender Reihenfolge sortiert) */
5  #include <stdlib.h>
   #include <math.h>
   #include <stdio.h>

10  int qsort_icmp(int*,int*);
   int qsort_icmp(x,y)
   int *x, *y;
   {
   if (*x<*y)    { return ( -1 ); }
   else if (*x==*y) { return ( 0 ); }
15  else        { return ( 1 ); }
   }

   int main(argc,argv)
   int argc;
   char *argv[];
20  {
   int pnt[11][100000];
   int qs[100000];
   char *dname;
25  int frac,i;
   long lauf,lower_bound,upper_bound;
   long l,no_pnt,seed_run;
   double ctime[11],x;
   FILE *datei;
30

   if (argc<3) {
   printf("\n\nZuwenig Argumente! \n\n"); printf("Aufruf: %s <Dateiname> <Konfidenzniveau (in
   %%>\n\n", argv[0]); return 1;
35  }

   dname = argv[1];
   frac = 100-atoi(argv[2]);
   lauf = 0;
40

   datei = fopen(dname,"r");
   if (datei==NULL) { return 1; }
   else {
   while (!feof(datei)) {
45  lauf++;
   for (i=1;i<=9;i++) {
   fscanf(datei,"%8lf %6d ", &ctime[i], &pnt[i][lauf]);
   }
   fscanf(datei,"%8lf %6d \n", &ctime[10], &pnt[10][lauf]);
50  }
   fclose(datei);
   }

   lower_bound = (long)floor(lauf*frac/200.); upper_bound = (long)ceil(lauf*(200.-frac)/200.);
55  if (lower_bound<1) {lower_bound = 1;}

```

```
printf("\n\n%2d%%-Sicherheitsbereich bei %d Simulationsläufen\n\n",
100-frac,lauf);
for (i=1;i< ;i++) {
5  for (l=1;l<=lauf;l++) {
    qs[l] = pnt[i][l];
  }

  qsort(&qs[1], lauf, sizeof(int), &qsort_icmp);
10 printf("Zeitpunkt: %8.2f Fehlerintervall: [%d,%d]\n",
    ctime[i], qs[lower_bound], qs[upper_bound]);
  }

  return 0 ;
15 }
```

Literaturverzeichnis:

- [1] Sidney I. Resnick: "Adventures in Stochastic Processes",
Birkhäuser Boston, 1992, ISBN 3-7643-3591-2, pp.303-317.
- [2] Bratelly et al., 1987

Patentansprüche

1. Verfahren zur Vorhersage von Meßdaten bis zu einem
Endzeitpunkt anhand vorgegebener Meßdaten,
 - a) bei dem ein stochastischer Prozeß an die vorgegebenen
Meßdaten angepaßt wird;
 - b) bei dem ab einem vorgegebenen Zeitpunkt
Simulationsläufe des stochastischen Prozesses bis zu
dem Endzeitpunkt durchgeführt werden;
 - c) bei dem für jeden Simulationslauf die prognostizierten
Meßdaten bestimmt werden;
 - d) bei dem die Vorhersage von Meßdaten durch Angabe eines
Wertebereichs, der durch die prognostizierten Meßdaten
bestimmt ist, erfolgt.
2. Verfahren nach Anspruch 1,
bei dem ein Konfidenzintervall für die Vorhersage von
Meßdaten bestimmt wird, indem die a% kleinsten und die b%
größten prognostizierten Meßdaten eliminiert werden.
3. Verfahren nach Anspruch 2,
bei dem a% gleich b% ist.
4. Verfahren nach einem der vorhergehenden Ansprüche,
bei dem der stochastische Prozeß ein nichthomogener
Poisson-Prozeß ist.
5. Verfahren nach einem der vorhergehenden Ansprüche,
bei dem die Meßdaten Anzahlen von Fehlern sind.
6. Verfahren zur Vorhersage von Meßdaten anhand vorgegebener
Meßdaten,
 - a) bei dem ein stochastischer Prozeß an die vorgegebenen
Meßdaten angepaßt wird;
 - b) bei dem ein Intervall ermittelt wird, indem anhand des
stochastischen Prozesses gewonnene

Wahrscheinlichkeitswerte der Größe nach um einen Erwartungswert sortiert werden;

- c) bei dem die Vorhersage von Meßdaten innerhalb der Grenzen des Intervalls erfolgt.

5

7. Verfahren nach Anspruch 6, bei dem die anhand des stochastischen Prozesses gewonnenen Wahrscheinlichkeitswerte der Größe nach symmetrisch um den Erwartungswert sortiert werden.

10

8. Anordnung zur Vorhersage von Meßdaten bis zu einem Endzeitpunkt anhand vorgegebener Meßdaten, bei der eine Prozessoreinheit vorgesehen ist, die derart eingerichtet ist, daß

15

- a) ein stochastischer Prozeß an die vorgegebenen Meßdaten anpaßbar ist;

- b) ab einem vorgegebenen Zeitpunkt Simulationsläufe des stochastischen Prozesses bis zu dem Endzeitpunkt durchführbar sind;

20

- c) für jeden Simulationslauf die prognostizierten Meßdaten ermittelbar sind;

- d) die Vorhersage von Meßdaten durch Angabe eines Wertebereichs, der durch die prognostizierten Meßdaten bestimmt ist, erfolgt.

25

9. Anordnung zur Vorhersage von Meßdaten anhand vorgegebener Meßdaten,

30

- bei der eine Prozessoreinheit vorgesehen ist, die derart eingerichtet ist, daß

- a) ein stochastischer Prozeß an die vorgegebenen Meßdaten anpaßbar ist;

- b) ein Intervall ermittelbar ist, indem anhand des stochastischen Prozesses gewonnene Wahrscheinlichkeitswerte der Größe nach um einen Erwartungswert sortiert werden;

35

- c) die Vorhersage von Meßdaten innerhalb der Grenzen des Intervalls erfolgt.

Zusammenfassung

Verfahren und Anordnung zur Vorhersage von Meßdaten anhand vorgegebener Meßdaten

5

Es werden ein Verfahren und eine Anordnung zur Vorhersage von Meßdaten anhand vorgegebener Meßdaten angegeben, bei dem/bei der ein stochastischer Prozeß an die vorgegebenen Meßdaten angepaßt wird. Ab einem vorgegebenen Zeitpunkt werden

10

Simulationsläufe bis zu einem Endzeitpunkt durchgeführt. Für jeden Simulationslauf werden die prognostizierten Meßdaten bestimmt. Für den Endzeitpunkt erfolgt die Vorhersage von Meßdaten innerhalb eines Wertebereichs, der durch die prognostizierten Meßdaten bestimmt ist.

15

98 P 5929

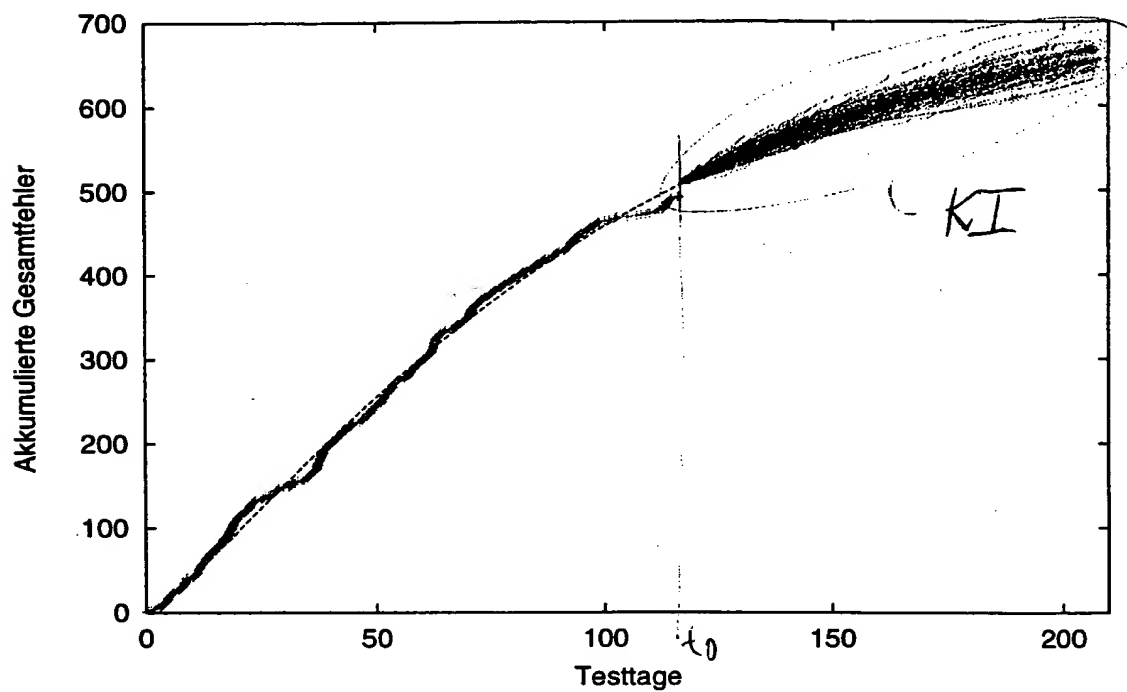


FIG 1

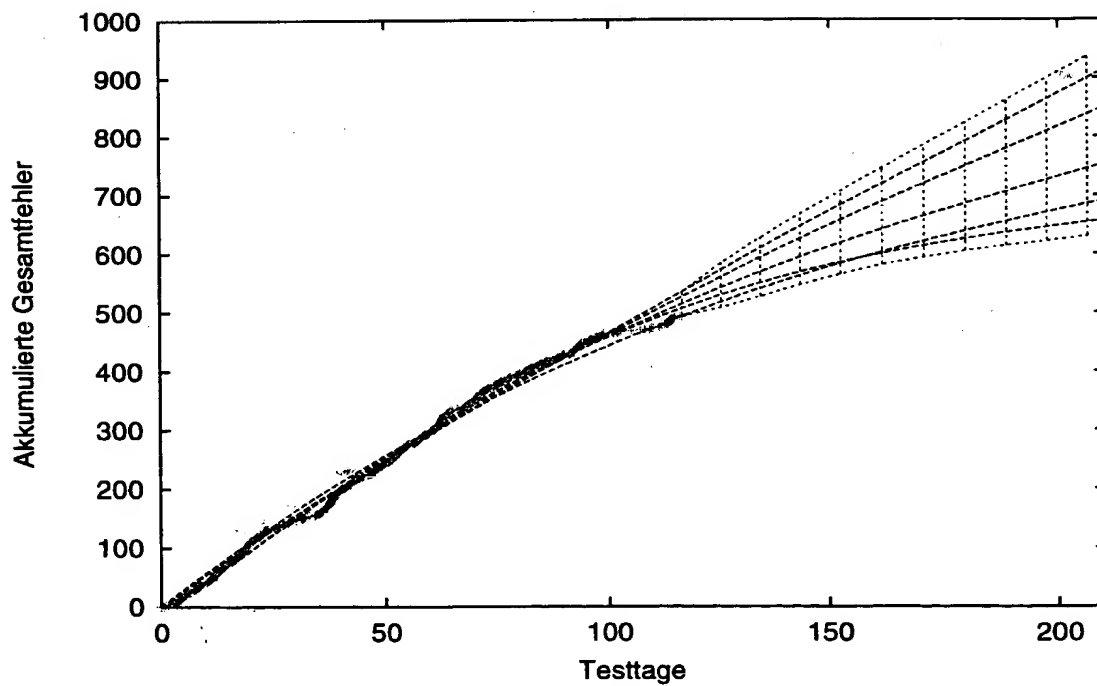


FIG 2

FIG 3

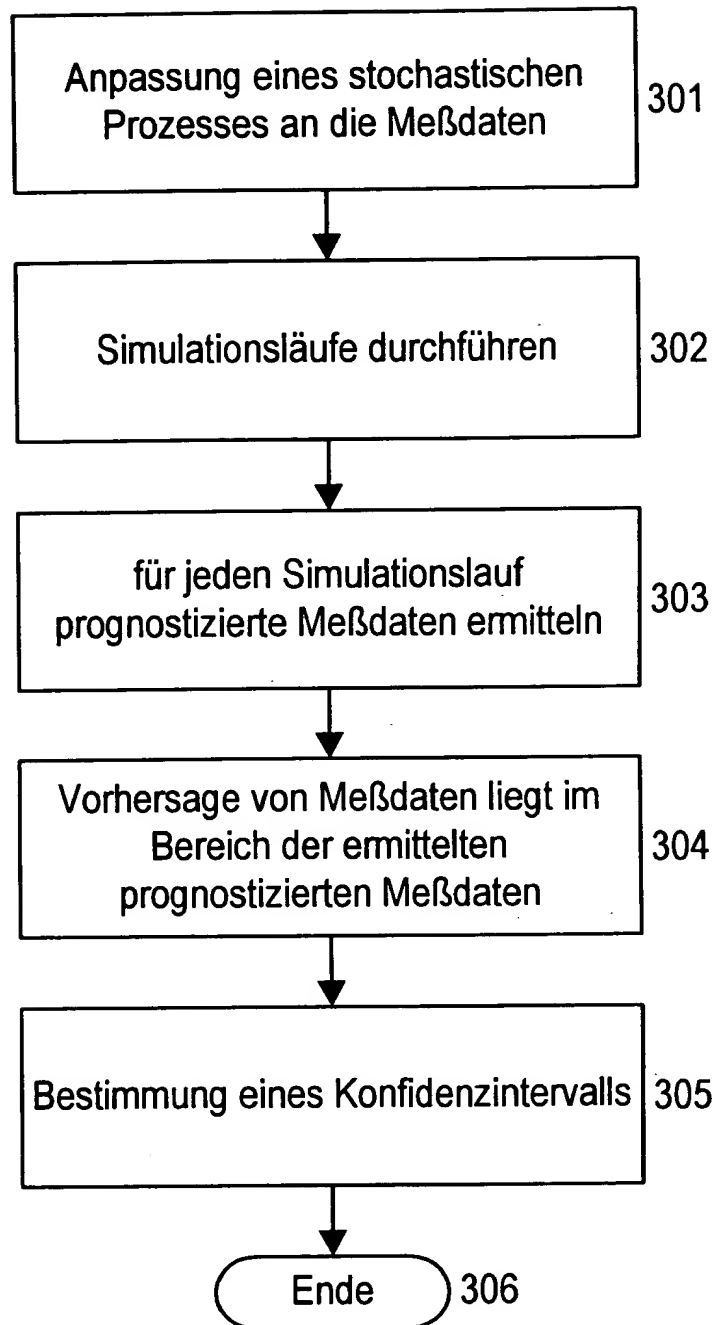


FIG 4

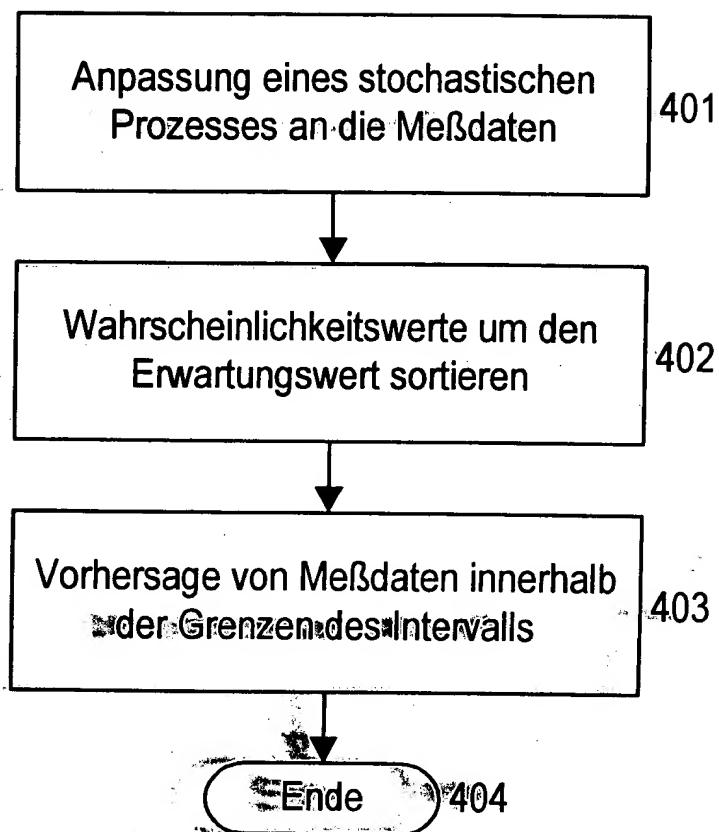
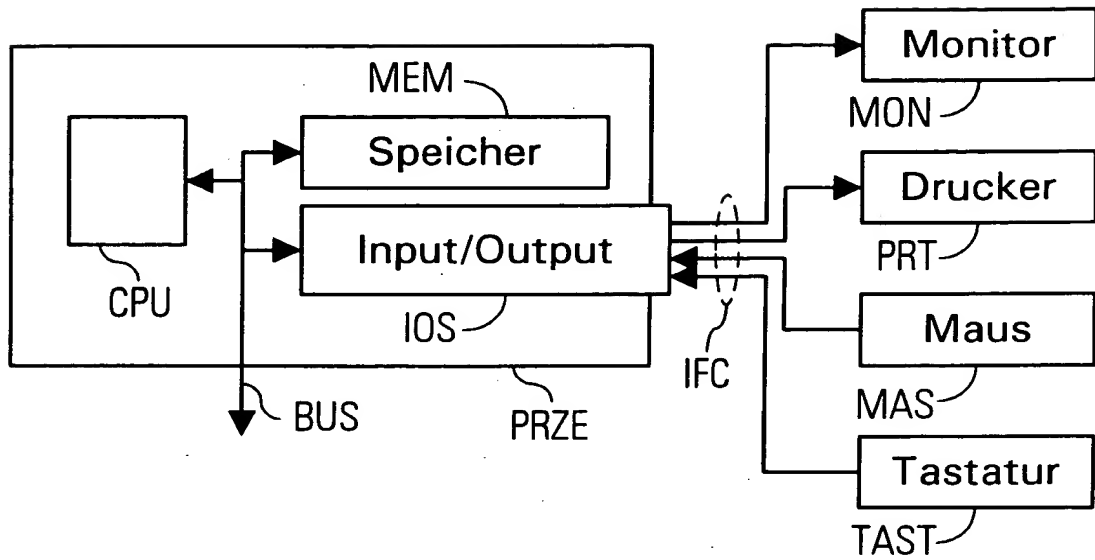


FIG 5



THIS PAGE BLANK (USPTO)